REMARKS/ARGUMENTS

Claims 6, 12, and 18

Claim 6 includes "adding a termination point to the critical section if a portion of the critical section is outside of the dependency graph". The Office Action (OA) rejects this based on Li. OA, 8. Specifically, the OA rejection rationale consists entirely of the following "Fig. 3B, item 328 and related text". *Id.* One may argue adding the ADVANCE instruction 328 of Figure 3B constitutes "adding a termination point to the critical section." However, what about the rest of the claim—"if a portion of the critical section is outside of the dependency graph." It appears any critical section of Figure 3B is included entirely within the dependency graph (DG) of Figure 3B. After all, Li says that AWAIT and ADVANCE instructions mark the boundaries of critical sections. *See* ¶¶35, 44, and Figure 5 (block 510). If that's true, and Figure 3B clearly has both instructions, then Figure 3B has a complete critical section and fails to cover a "portion of the critical section is outside of the dependency graph." Simply put, Li just doesn't talk about what to do when critical sections extend outside DGs. It just isn't Li's focus or concern. Please reconsider the rejection. **If the rejection is maintained, please identify:**

- a specific DG,
- a specific critical section that has a portion extending outside the DG, and
- a specific termination point added to the critical section (that extends outside the DG).

Applicant posits no such identification is possible in Li.

For at least these same reasons claims 12 and 18 are also allowable.

New claim 20 is supported in the least at ¶¶ 33-34, 50-55 of Applicant's published specification.

Claims 7, 15, and 19

Claim 7 includes "inserting additional dependency relationship based on a direct dependency, an indirect dependency, or a shortest life-time dependency." The OA rejects this based on Lauterbach, which allegedly inserts "artificial dependencies into the generated dependency graph." However, the OA fails to explain how "artificial dependencies" are being matched to the specific types of dependencies of claim 7. The BPAI held a post-KSR obviousness determination requires the PTO make "**a searching comparison of the claimed invention–including all its limitations–with the teaching of the prior art**." *In re Wada and*

Murphy, Appeal 2007 3733 (2008)(Fed. Cir. citations omitted). As such the present OA rejection rationale does not create a prima facie 103 case for claim 7. Please reconsider the rejection.

To expedite prosecution, claims 15 and 19 have been amended to further focus on particular types of dependencies. If the amended claims are rejected, please specifically address the particularly claimed dependencies.

Claims 1, 15, and 19

Claim 1 includes inserting dependency relationships "based on the dependency graph between the plurality of blocks to cause execution of the detected portion of the computer program outside of the critical section." The OA rejects this based on Li. Applicant respectfully requests clarification for the rejection. Please specifically identify these elements of claim 1:

- The "dependency graph",
- How the DG is "between the plurality of blocks" (i.e., identify the "blocks"),
- The "dependency relationships", and
- How the dependency relationships are inserted "based on" the DG.

Doing so will illustrate a shortcoming of Li. Specifically, Li provides:

> In other words, the detection and sinking of sink instructions, as well as hoist instructions, should not violate any data dependencies or, for example, control dependencies, indicated by a dependence graph of the sequential application program. In other words, compliance with a dependence graph ensures that program-threads generated from the sequential application program. In one embodiment, program-threads maintain sequential semantics of the original program and enforce dependencies between program-thread iterations corresponding to a PPS loop of the sequential application program.

¶64. In other words, Li appears to create a DG based on the original program. It then works to split the infinite loop up for separate thread execution. Li "enforce[s] dependencies" along these lines. But claim 1 asks the Examiner to focus on DG aspects. In other words, claim 1 discusses how a specific DG is constructed and manipulated. At best, Li shows a DG get constructed (Figure 6A) and then reduced to focus on a critical section (Figure 6B). But the relationships in Figure 6B were already present in Figure 6A. **So, Li just streamlines Figure 6B—it does not "insert[] a plurality of dependency relationships" into the DG—it just removes the**

relationships from Figure 6B. It's the opposite. And along these same lines, Li certainly does not insert the dependency relationships "based on" the dependency graph.

For at least these same reasons, claims 9, 13, and 16 area allowable.

New claims 21 and 22 are supported in the least at ¶55 of Applicant's published specification.

Conclusion

The application is believed to be in condition for allowance and the Examiner's prompt action in accordance therewith is respectfully requested. The Commissioner is authorized to charge any additional fees or credit any overpayment to Deposit Account No. 20-1504.

Respectfully submitted,

Date: May 16, 2011

/E.E. "Jack" Richards, II/
E.E. "Jack" Richards, II
Registration No. 53,514
TROP, PRUNER & HU, P.C.
1616 S. Voss Road, Suite 750
Houston, Texas 77057-2631
(512) 418-9944 [Phone]
(713) 468-8883 [Fax]
Customer No.: 21906